

[19] 中华人民共和国国家知识产权局

[51] Int. Cl⁷

G06F 1/00

[12] 发明专利申请公开说明书

[21] 申请号 99803708.7

[43] 公开日 2001 年 4 月 25 日

[11] 公开号 CN 1292897A

[22] 申请日 1999.11.1 [21] 申请号 99803708.7

[30] 优先权

[32] 1998.11.10 [33] US [31] 09/189,534

[86] 国际申请 PCT/EP99/08333 1999.11.1

[87] 国际公布 WO00/28398 英 2000.5.18

[85] 进入国家阶段日期 2000.9.5

[71] 申请人 皇家飞利浦电子有限公司

地址 荷兰艾恩德霍芬

[72] 发明人 Y·E·什特伊恩

[74] 专利代理机构 中国专利代理(香港)有限公司

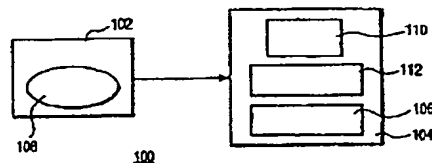
代理人 马铁良 张志醒

权利要求书 2 页 说明书 6 页 附图页数 2 页

[54] 发明名称 用于版权保护的做为软件对象提供的内容

[57] 摘要

诸如电影、音频文本或文本消息等内容信息以软件对象形式提供给终端用户,该软件对象具有一封装的过程用于终端用户在一运行时间环境中访问所述内容信息。所述对象能够规定时间帧,其中内容信息要被访问。由于所述过程与内容数据一起封装在对象中,还由于对象经互联网的输送是在串行化之后,因此提供一定程度的安全性,防止未授权的播放和复制。



ISSN 1008-4274

知识产权出版社出版



权 利 要 求 书

1. 一种把软件对象(108)的内容信息(110)提供给终端使用者的方法,其特征在于,所述软件对象具有一个在运行环境(104)中用于终端用户访问所述信息的已封装过程。
- 5 2. 根据权利要求1的方法,其中所述过程启动对内容信息的条件性访问。
3. 根据权利要求1的方法,其中还包括包含串行化所述软件对象的步骤,用来通过网络(208)电子地提供所述软件对象。
4. 根据权利要求1的方法,其中所述对象包含一个在播放设
10 备(104)上启动内容信息的播放的过程。
5. 根据权利要求1的方法,其中所述对象包含一个在存储器(204;206)中启动内容信息的存储的过程。
6. 根据权利要求1的方法,其中所述提供被包括在电子信函服务中。
- 15 7. 根据权利要求1的方法,其中所述提供被包括在电信会议服务中。
8. 根据权利要求1的方法,其中所述提供被包括于视频会议服务中。
9. 根据权利要求1的方法,其中所述内容信息包含音频信
20 息。
10. 根据权利要求1的方法,其中所述内容信息包含视频信息。
11. 根据权利要求1的方法,其中所述内容包含电影。
12. 一种接收一个或多个软件对象的内容信息的方法,其特征
25 在于,包括在运行环境(104)中的启动终端用户来访问内容信息的过程。
13. 根据权利要求12的方法,其中包括通过在所述对象中的条件性访问过程来启动对所述内容信息的条件性访问。
14. 根据权利要求12的方法,其中所述内容信息包括音频信
30 息。
15. 根据权利要求12的方法,其中所述内容信息包括视频信息。

00:09:05

16. 根据权利要求12的方法，其中所述内容信息包括电影。

17. 根据权利要求12的方法，其中所述内容信息包括字母数字信息。

18. 一种为终端用户制备内容信息的方法，其特征在于，该方法包括在一软件对象中对内容信息(110)打包，所述软件对象已封装了一个在运行环境(104)中的启动所述终端用户以访问所述信息的过程。

19. 根据权利要求18的方法，其中所述内容信息包括电影。

20. 根据权利要求18的方法，其中所述内容信息包括有字母数字信息。



说明书

用于版权保护的做为软件对象提供的内容

本发明涉及一种对内容数据提供可控访问的方法，特别是该内容数据被提供到用户电子设备的终端用户或多媒体计算机时的一种方法。本发明还涉及一种保护对数字数据内容免于未授权复制的方法。

这里所说的术语“内容”和“内容数据”是指那些含有终端使用者所用信息的数据，例如：字母数字文本、图形、视频、音频、多媒体等等。

一般来说，数字内容通常通过电缆、互联网或写入 DVD (数字化视频光盘) 等途径，从远程服务器被发送或流入到使用者终端。该内容经接收装置 (如：计算机、置顶盒) 处理并绘制给终端使用者。通过使用水位标的方法可对内容进行版权保护：即添加不能为用户觉察、但可被绘制 (rendering) 或校验软硬件所识别的数字式修改。另一种保护内容的方法是加密。通过使用一种特殊的加密算法使内容数据在发送到其目的地前被编码。并使数据流在绘制或存储之前在终端设备或在中间处理器 (PC、缆盒、置顶盒等) 上被解密。

以上这些方法尚存许多问题。例如，水位标对内容的修改是高度敏感的。如果数字内容被软件/硬件所修改，但并未感觉到水位标的存在，则水位标不能保证仍维持原初始形式。举例来说，调整图片大小、以不同的文件格式保存图片、应用压缩技术等也会影响水位标并降低其有用效果。加密方法也面临许多挑战。随着计算机/处理硬件和软件的更加强大和完善，即便是著名的运算法则也很快变得过时而被淘汰。另一个挑战是标准化。即，为了占据具有多样卖方的大规模市场，内容保护方法必须是标准的。但另一方面，标准的加密方法会在相对较短的时间里变得陈旧并被专家破译，因而也就很容易失效。

本发明的一个目的是提供一种对内容数据提供可控访问的替代方法。本发明的另一个目的是使数字内容更难于被未授权的代理商复制或绘制。本发明的第三个目的是提供一个安全且迅速兼容的方法对数字内容进行分配。

为达以上目的，本发明提供了一种提供或接收内容信息，特别是经由网络，作为一个或多个软件对象的方法。在运行环境中，过程被

封装在信息访问的对象中。也就是说，终端使用者访问信息的过程将与内容信息一起打包被传递给终端使用者。

我们知道，一个软件对象包含有内容数据及一个处理数据的方法或过程。在面向对象的编程中，方法就是一个已编程的过程
5 (PROCEDURE)，此过程被定义为一个类的一部分且被包含于该类的任何一个对象中。一个对象能够具有多个方法。对象中的一个方法只可访问该对象的已知数据。这样就保证了在应用中，数据在此组对象中的完整性。

开放的分布式软件结构如 HAVi、Java/JINI、基于 COM 或 DCOM
10 技术的 Home API、CORBA 等都允许在网络上（因特网、家庭网、局域网、广域网等）对软件对象进行传递转换。关于软件表示的更多信息，HAVi，应用 COM 技术和 OLE 自控对象，下面的专利文件在此结合引用为参考：U.S. 专利序列号 No. 08/731, 624（代理案号：PHA 23, 169），申请日 10/15/96，发明人：Paul Chambers 与 Saurabh Srivastava，
15 发明名称：任务驱动分布式多媒体用户系统；U.S. 专利序列号 No. 09/146, 020（代理案号：PHA 23, 492），申请日 9/2/98，发明人：Yevgeniy Shteyn，发明名称：低数据速率网络于高数据速率 HAVi 网络上的表示；U.S. 专利序列号 No. 09/165, 683（代理案号：PHA 23, 483），申请日 10/2/98，发明人：Yevgeniy Shteyn，发明名称：
20 经由有权路径调用控制软件对象的识别脚本；U.S. 专利序列号 No. 09/165, 682（代理案号：PHA 23, 484），申请日 10/2/98，发明人：Yevgeniy Shteyn，发明名称：将控制特性映射到形式兼容的 GUI 元件上；以及 U.S. 专利序列号 No. 09/107, 525（代理人案号：PHA 23, 438），申请日 6/30/98，发明人：Yevgeniy Shteyn 与 Gregory
25 Gewickey，发明名称：多重通讯协议系统中设备的动态注销。

当软件对象等被转移到机器上，它就可与机器的系统软件交互。举例来说，一个 HAVi DCM 可被上载到一个 FAV 上，并在 JAVA 的运行环境中被执行。DCM 可被编程到 API 工具实现绘制、存储、记录或是其他关于图形、视频、音频、文本、图象等的处理。在这种情况下，
30 软件对象就可直接与绘制设备工作而不须中间解密或修改。所有操作都被对象本身所控制并可对不友好的应用隐藏。当内容须要存储时，一个对象控制存储 API 就被调用（例如：串行化，在下面可看到）。

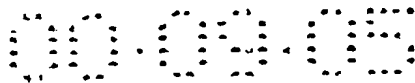
由于此对象控制着存储（记录）处理，主机系统就必须重建一个对象来访问已存储的内容，因此其内容就得到保护了。对象可影响记录的质量、内容可被记录或绘制的次数以及其他与主机系统的交互。软件对象也可被做成时间敏感的。例如：一个电影对象可被租用，也就是，
5 其绘制 API 只可在一个规定的时间长度—电影播放时间内被使用，如一天，等等.... 一个控制对象不需要包含所有的内容。当内容很大时，对象可只保留对内容位置的引用并在需要的时候进行检索。在必须或适合的时候，对象间的链接可以控制操作。例如，不同的电影场景由不同的对象所控制。一个对象可被设置为对内容的某些部分提供免费的（轨迹）访问，而对其他部分则要求付费。与系统资源有条件的交互可以基于用户的优先选择而被实现。广告和促销可作为独立的对象被插入是依赖于用户访问的级别。（举例来说，通过预约）。

我们知道，一个对象可隐藏二进制码，如果对象客户的运行环境已知或对象与设备交互后可被检测的话。例如，一个较好的绘制算法
15 或二进制升级可被传递到一个运行 Java 虚拟机的三媒体机器上。

此种被推荐的通过对象将数据内容传递到一个终端的方法，除了适合版权保护外对于其他使用也是非常适合的。如：email(电子信函)、电信会议（经由电话）或视频会议出于安全因素可以软件对象的交换为基础。类似地，电子货币交易也可通过软件对象的通讯而安全地进行。电子信函也可作为软件对象被实现。

对象通过网络的传输，例如从服务器端到客户端，是通过如对象串行化等来实现的。Java 中的对象串行化支持各对象的编码进入比特流，且对象对其是可达的，它还支持源于该流的对象图形的互补性重建。串行化用作轻度持续以及用于经由插座或远程方法调用（RMI）
25 通讯。默认的对象编码保护个人的和瞬时的数据，并且支持类的进化。一个独立的数字数据流可被在传统意义上不存在的用户恶意盗版并复制。所有被发送的都是比特串，并且由于过程和内容信息是不可能从流中被识别的，所以对大部分人来说不可能找到其意义的线索。一个类可以实现其自身的外部编码来增加安全性，然后单独负责其外
30 部 格 式

（source:<http://www.javasoft.com/products/jdk/1.1/docs/guide/serialization/index.html>）。因此，一个解密过程可与加密的



内容数据一起被封装于对象中。该解密过程就在接收结束时被调用，可能还被一个封装的确认接收者身份、解密密钥或绘制设备序列号等的检测程序所补充。

为了清晰起见，我们知道经由因特网可以下载 Applet，对 Applet 的用户访问需要一个驻留在接收客户端的浏览器应用程序。该 Applet 在浏览器应用程序中被执行。浏览器或插件程序不是该 Applet 的组成部分。在此发明中，软件对象在接收机运行环境（举例来说，HAVi 运行时间，Home API 运行时间 [=Windows OS]，Jini 运行时间）中使用其自身程序直接激活，并且在运行时间中不需要浏览器或任何客户应用程序。

本发明的各种有利的方案都被列在所附的权利要求及其所有可能的组合中。例如，一个电视会议系统可提供常规的视频流以及对象封装的安全音频和/或图形数据。

参考附图借助实例更详细地对本发明进一步说明，其中：
图 1-5 是本发明的方法得到执行一个系统实例的方块图。
图中相同的数字参考号表明相似或相应的特征。

图 1 是一个执行本发明方法的 HAVi 系统 100 的方块图。系统 100 包含一个存储媒体 102 和一个 FAV（完整 AV 节点）104，在本例中 FAV 包含一个数字电视。FAV 104 有一个 Java 虚拟机 106。存储媒体 102 包含（如）一个缓冲器，它存储从外部系统 100（如）经去串行化后的因特网或 DVD 或其他包含软件对象的物理数据载体以电子方式接收的软件对象 108，这里是 HAVi DCM 108 具有内容数据 110 以及由绘制 API 112 表示的播放程序，。软件对象 108 被上载在 FAV 104 中的绘制处理，在 FAV 104 中绘制 API 104 与 Java 虚拟机 106 一起直接交互作用。对内容的有条件访问可基于安全性或用户/系统容量。在基于安全的模式中，对象允许通过 API 接口对于对检验有贡献的播放或存储调用。运行时间应必须提供一个可被对象检验的权限或标识符。例如，父本（parental）控制可由代表电影不同部分内容的一组对象来实现。作为另一个例子，一本电子书籍可以封装带有限制性要求访问的图形对象。

图 2 是一个使用 JavaSpaces 结构的 Jini 系统 200 的方块图。JavaSpaces 与 Java 虚拟机和 Java 远程方法调用的功能性一样，其结构用于建立一个分布操作系统及建立基于 Java 对象的存储库。JavaSpaces 的执行提供了一个机制，用来存储相关对象组并基于对指定域查询匹配值以对此对象组进行检索。这允许 JavaSpaces 服务器被用来在远程系统上存储和检索对象。JavaSpaces API 使用一种特殊的包来提供基本的原子处理，成组多重操作越过多重 JavaSpaces 执行成为作为单原子操作的束。JavaSpaces 结构支持机制处理，允许多操作和/或多空间的更新来完成。术语“原子性”的意思是按某种处理所有操作成组出现或者一个都不出现。Jini 也是一种源于 Sun Microsystems 的对于网络设备的技术。Jini 是一种基于 Java 的软件技术，辅助个人电脑网络和外围设备。当被插入网络时，Jini 使能设备将播报其存在。准备使用此设备的网络客户可从设备本身要求必要的软件，忽略服务器或网管。这种结构是建立于一个已存在的网络的顶端的。

系统 200 包含 JavaSpaces 服务器 202 和 204，以及通过网络 208 连接的客户端 206。JavaSpaces 202 或 204 提供一个已分配的持续性和写在 Java 程序语言中的对象互换机制。对象在入口处被写入，该入口提供一个对于相关域的已键入的分组。客户可在 JavaSpaces 服务器上执行简单的操作来写入新的入口、查找现存入口以及从空间中删除入口。用户使用这些工具可写入系统，该系统使用数据流来实现分布运算法则；并可使 JavaSpaces 系统实现对用户的持续性，当有一个适当的请求时，服务器 202 将一个对象转移（通过串行化）至客户端 206。此转移就如 Java 的分配租用清单所指定的一样是一种租用。除非对服务器或资源的授权被授权者明确地取消，不会对服务器或资源授权，一个已租用的资源或服务器授权是基于时间的。当租用期满时服务器将终止或资源被释放。“租用”的概念在分布系统中是特别有用的，在这种情况下协作组的不同部分都是不可通信的，举例来说，由于组或连接成员的失败。根据图 2 的图表，服务器 202 将一个对象租给服务器 204 作临时存储并将次对象租给客户端 206 作临时存储或绘制。这种布局是与如视频（或音频）点播服务商、多玩家视频游戏供应商等有关的。相似的客户端-服务器端结构可使用微软

的基于 Windows 操作系统并附带一个 COM 客户端的 COM/DCOM 技术来建立。

图 3 表示系统 300, 它具有 COM/DCOM 服务器 302 和 COM 客户 304, 服务器 302 提供对象访问和对象存储。客户端 304 可使用 COM 或 DCOM 机制访问服务器 302。在远程位置时使用 DCOM。访问机制对客户端 304 是透明的。COM 对象被转移到客户端 304 进行处理, 并且一个适当的 API 在客户端 304 被调用其目的是产生所希望的内容。对象对系统资源和 API 有完整访问, 如微软的 DirectShow 组件。

图 4 是本发明的另一个系统 400 的配置图。系统 400 包括一个服务器 402, 用于提供对象并充当对象存储库。系统 400 还包含客户端 404 和 406。系统 400 用于 (例如) 内容交换的多客户交互, 比如视频会议。内容对象被客户端 404 放置在服务器/存储库 402 中, 并且客户端 406 可从存储库 402 中对其进行检索。

图 5 是与系统 400 配置相似的系统 500 的配置图。系统 500 包括一个服务于客户端 504、506、508 和 510 的对象服务器 502。服务器 502 的功能是作为一对对象存储库来使能多客户端对象交互作用。内容对象可被 504-510 的任何客户端放置于服务器 502 中并可被除客户端 504-510 的任何客户端所检索。系统 500 包括一个中继或复制服务器 512 以服务于客户端 514 和 516。一个内容对象可被打包或根据客户端 504-510 和 514-516 优先权或访问特权而暴露界面。

关于更多的信息, 相关的 Java 的说明书、JavaSpaces 和 Jini 结构、所有 Sun Microsystem, 以及微软的全部组件对象模型说明书, 可以作为参考,

说明书附图

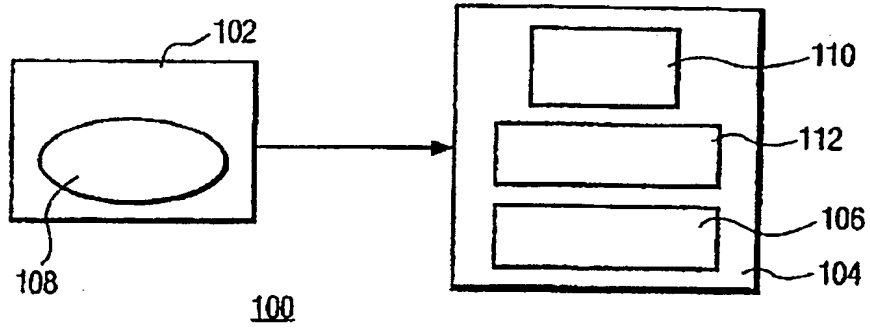


图 1

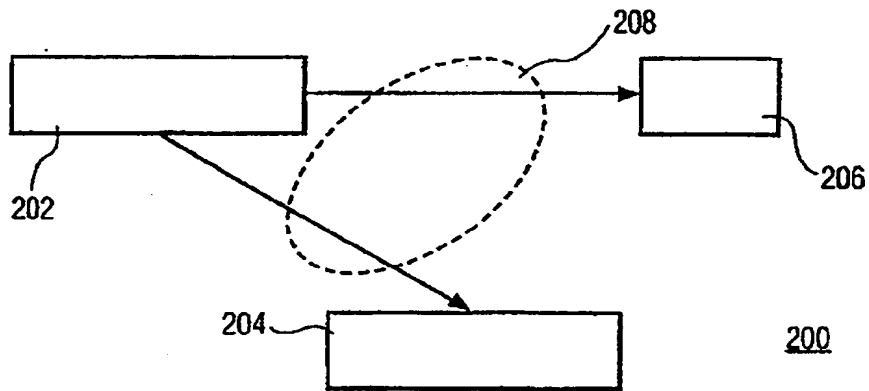


图 2

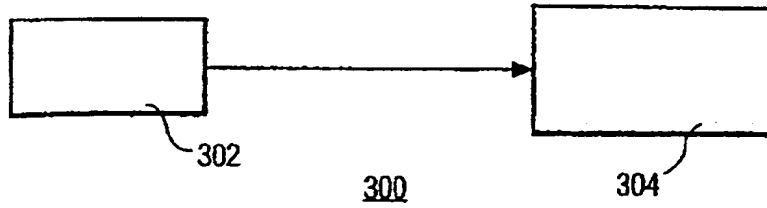


图 3

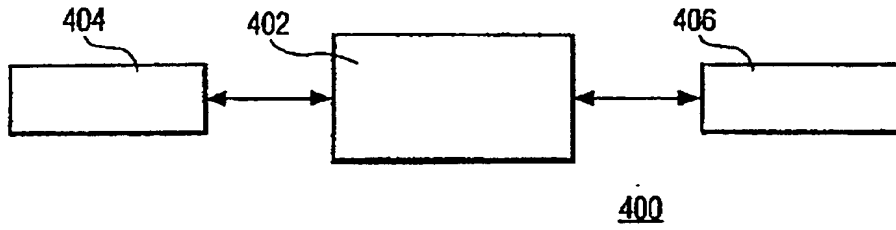


图 4

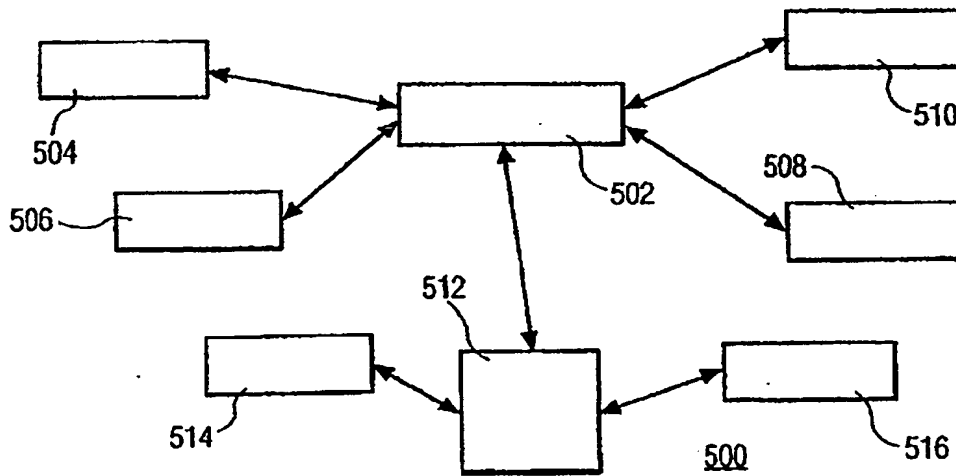


图 5

Concise Explanation

on a reference "Introduction to Java Programming in i-mode",
published on March 26, 2001, pages 30 to 41.

The first part, pages 30 to 36, describes general concepts of Java. The first part also includes a note titled "Is i Appli an application program?" in page 35. The note states that an embodiment discussed in this document of executing Java programs in compliance with i-mode is referred as i Appli.

The second part, pages 37 to 41, describes how Java is implemented in a mobile phone. The rest of this document includes partial English translation of the second part

[page 37]

(1) User's view of Java implementation in a mobile phone

In i-mode, for example, a user of a mobile phone executes a Java application program which has already been downloaded by selecting the Java application program from a proprietary menu for executing Java application in a similar manner of selecting contents from i Menu in existing i mode.

(2) internal implementation

While some Java applications may be pre-installed in a mobile phone in some case, Java application is, in general, not installed in a mobile phone. Java execution environment for executing Java applications is only preloaded in a mobile phone. Then, a Java

application is installed via some network.

[page 39]

A Java application is placed in a server in combination with a file called “ADF (Application descriptive File)” simply including the outline of the Java application.

When an “ADF” of Java application that a user desires for the services is requested, a download page for the Java application is prepared and a link to a URL of the ADF is provided on the page. A user clicks the link

The outline of Java application such as the name of Java application to be executed, a file size, and a version are written in the ADF. When downloading of the file completes, a module called “Java Application Manager (JAM)” for managing Java application checks the contents of ADF. The Java Application Manager interprets the Jar application name and start downloading the Java application.

Java application is downloaded as a Jar type file which is a compressed format for distributing Java program. When the download completes, JAM checks the file size of the downloaded Jar to compare with that of ADF. When the size agrees, JAM stores the Jar file in application dedicated memory. In a case that the Java application having the same name is already installed, the installed file is updated to a newly acquired file after checking the date and the version of it.